2011

Delay Tolerant Networks (CSUMB Brief)

Cebrowski Institute

http://hdl.handle.net/10945/37265

Delay Tolerant Network Routing

Sathya Narayanan, Ph.D.

Computer Science and Information Technology Program

California State University, Monterey Bay

This work is supported by the Naval Postgraduate School – Military Wireless Communications Research Group

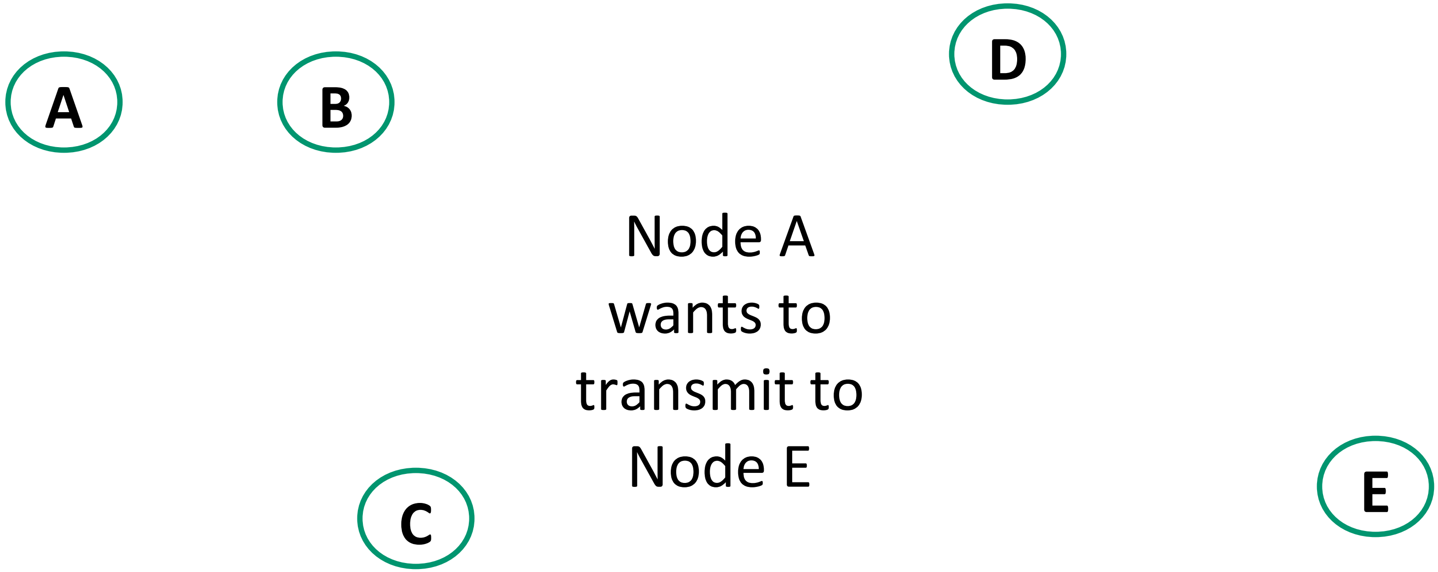
Overview of Talk

- **Background**
- **Research Objective**
  - Performance analysis
  - Message Prioritization
- **Simulation Study**
  - Results
- **Future Plans**

Delay Tolerant Network Routing

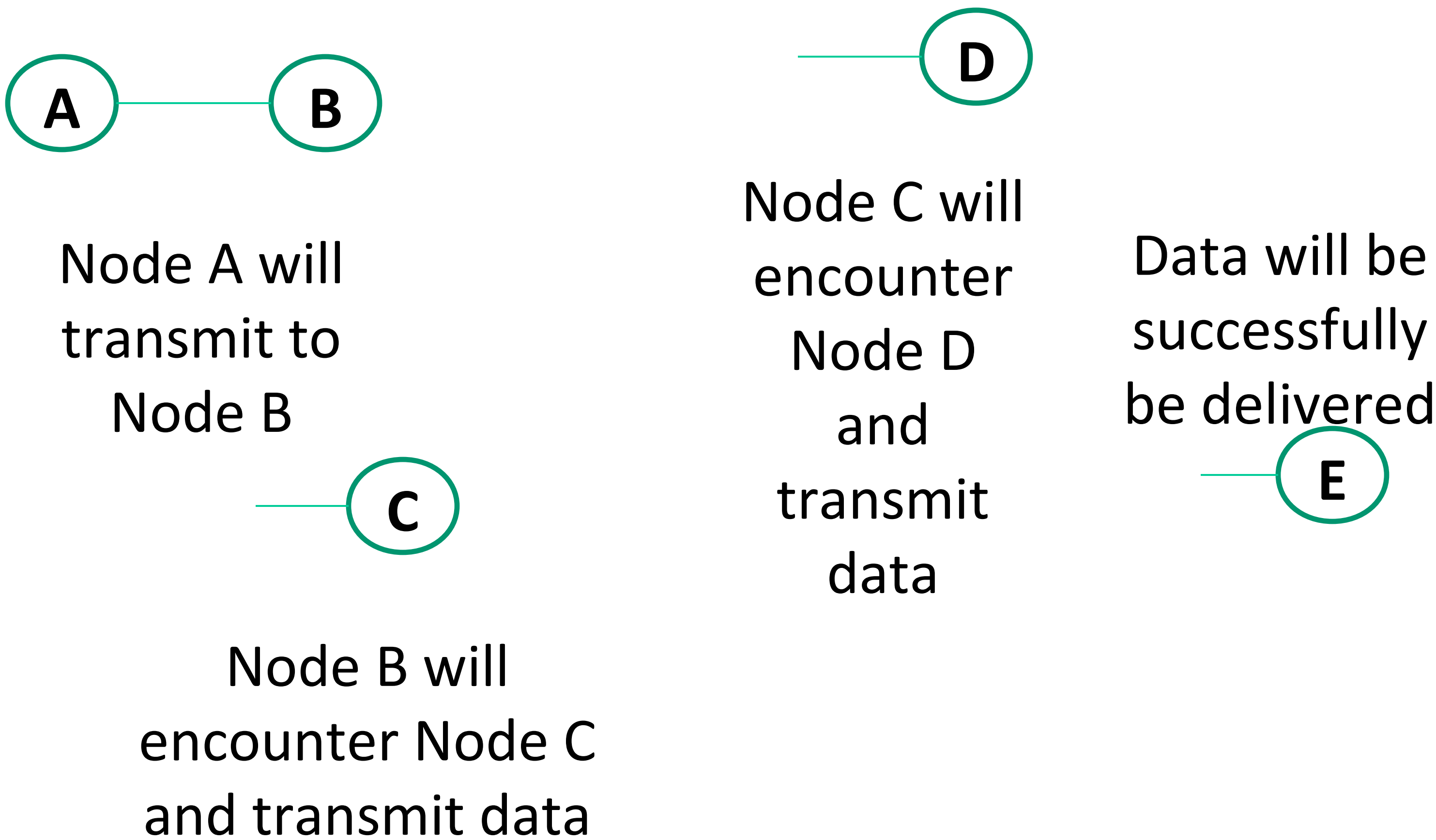
- **Traditional networks**
  - Route from source to destination exists when the message leaves the source
- **Delay tolerant networks**
  - No pre-existing route
  - Message is forwarded as nodes encounter each other
    - Message traverses the route over time as the nodes move around

Delay Tolerant Network



Node A wants to transmit to Node E

Delay Tolerant Network



Routing Protocols

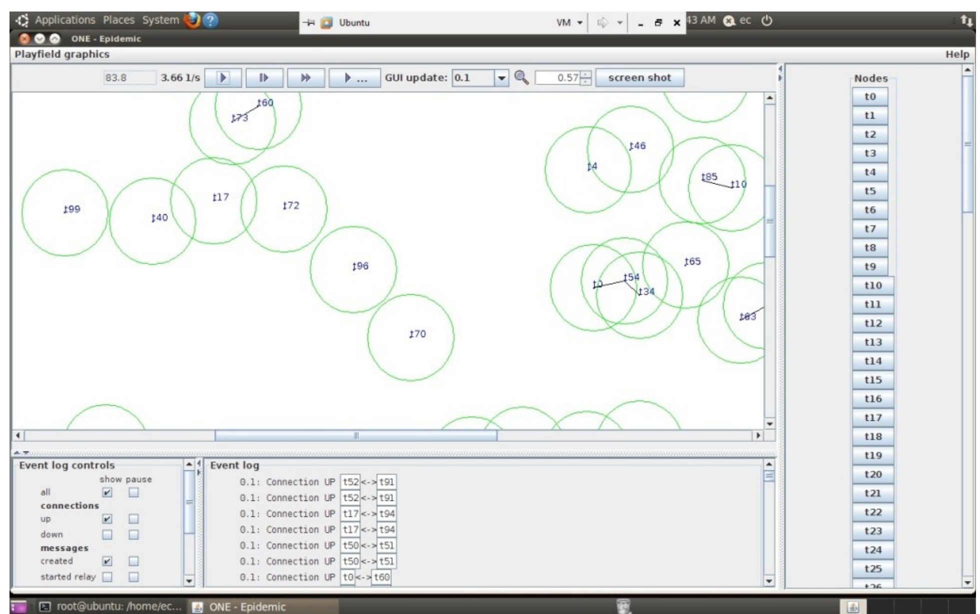
- **This research focuses on two routing protocols**
- **Epidemic Routing**
  - Forward message to every node encountered
  - Message spreads like that of a disease in a population
- **ProPHET**
  - Probabilistic Routing Protocol using History of Encounters and Transitivity
  - Use past encounters to predict future best route
  - Provides a framework allowing for different forwarding decision algorithms

Research Objective

- **Message Prioritization**
  - Use the insights gained from analysis to develop message prioritization algorithms for DTN routing
- **Performance analysis**
  - Develop analytical and simulation models to study three related performance parameters
    - Duplicate messages in the network at the time of delivery
    - End to end latency of message delivery
    - Probability of message delivery

Current Status

- Developed four types of ProPHET forwarding decision algorithms
- Developed a simple probabilistic extension to Epidemic (q – Epidemic)
- Extensive simulation analysis of Epidemic vs ProPHET routing using ONE (Opportunistic Network Environment Simulation tool)



Results

- **A lot of data collected**
- **Some insights:**
  - q = 0.5 Epidemic has similar performance as ProPHET without all the complexity when Random Waypoint Mobility is used
  - Aggressive algorithms have low latency at low message generation rates
  - We haven’t seen any consistent performance improvement by ProPHET when there is any randomness in the mobility pattern (More simulations are being run as we speak)

Results

- **Insights continued:**
  - Variables that impact the latency are:
    - Message generation rate
    - Queue length
    - Number of nodes
    - Aggressive vs non aggressive algorithms

Sample Results

10 Nodes														
			Epidemic			Epidemic			Prophet			Prophet		
			q0.5			q1.0			Type2			Type4		
	Src	Dst	Duplicates	Latency	Probability	Duplicates	Latency	Probability	Duplicates	Latency	Probability	Duplicates	Latency	Probability
Low	1	1	4.6	5854	0.9	8.2	2321	1	1.8	7164	0.4	3.2	8795	0.5
	1	2	4	5471	0.9	8.2	1919	1	2.1	7114	0.7	4.2	6232	0.9
	2	2	3.7	3601	1	7.6	1546	1	2.5	4283	1	3.4	3535	1
Med	1	1	3.4	9043	0.5	4.6	6951	0.9	2.19	8783	0.3	2.6	8837	0.4
	1	2	2.9	8522	0.7	4.4	6464	0.9	2	8321	0.6	3.5	7441	0.8
	2	2	2.7	6450	1	3.8	4273	1	2.5	6765	0.8	3.2	4975	0.9
High	1	1	3.3	8519	0.4	4.5	8721	0.6	2.13	10417	0.2	3.6	10579	0.2
	1	2	3	9007	0.5	4.3	8805	0.7	1.75	7737	0.4	2.8	8165	0.4
	2	2	2.7	9112	0.6	3.7	8236	0.8	2	7259	0.8	3	6764	0.8

40 Nodes														
			Epidemic			Epidemic			Prophet			Prophet		
			q0.5			q1.0			Type2			Type4		
	Src	Dst	Duplicates	Latency	Probability	Duplicates	Latency	Probability	Duplicates	Latency	Probability	Duplicates	Latency	Probability
Low	1	1	14.5	1389	1	22.5	724	1	3.2	6669	0.6	9.3	3277	1
	1	2	13.6	1368	1	22.7	744	1	3	5537	0.9	10.3	1665	1
	2	2	13.2	1097	1	20.3	663	1	3.4	2633	1	10.7	1258	1
Med	1	1	14	2218	1	21.9	793	1	2.7	7585	0.6	8.5	5057	0.8
	1	2	12.3	1681	1	21	800	1	2.7	6883	0.8	8.8	3078	1
	2	2	11	1550	1	20	714	1	3.1	3599	1	9.8	1572	1
High	1	1	10.3	5689	1	23.1	1759	1	3.2	8447	0.3	10.1	7506	0.5
	1	2	9.7	5862	0.9	19.5	1626	1	2.8	8693	0.6	8.1	5881	0.9
	2	2	7.9	3387	1	21.7	1068	1	2.7	6802	0.9	9.1	3982	1

Conclusion

- **Throttling Epidemic behavior using a q value seems to work well**
- **Mathematical analysis based on the input variables is needed**
  - Work in progress
- **Few levers available to affect message prioritization at routing**
  - q value for Epidemic
  - Limit on the number of hops
  - Prioritization within queues

Two Related Recent Projects

- **Experimentation with Simple Message Prioritization Extensions to ProPHET**
  - NPS Master Thesis (March 2011, LT Rapin, USN)

- **Secure Distributed Storage for Mobile Devices**
  - NPS Master thesis (March 2011, LT Huchton, USN)
  - Upcoming MILCOM paper

Experimentation with ProPHET Message Prioritization

- **Simple extensions (with two traffic priority classes) can increase the performance of high priority messages significantly**
  - Higher message delivery rate
  - Lower message latency
- **Urgent need of stable software prototypes to advance DTN research beyond theory and simulations**
  - The current IRTF DTN2 reference implementation is of very low quality

A Secure Distributed File System for Mobile Devices

- **Resistant to total device compromise**
  - Up to a customizable number (*k*) of device captures
  - No need for specialized tamper-resistant hardware
  - Addressing limitation of “Remote Kill”
- **Group secret sharing also supports data resiliency**
  - Different collection of *k* devices can recover data
- **Prototype on Android 2.2 Smart Phones**
  - write() and read() throughput performance: up to 15 Mbps

Backup Slides

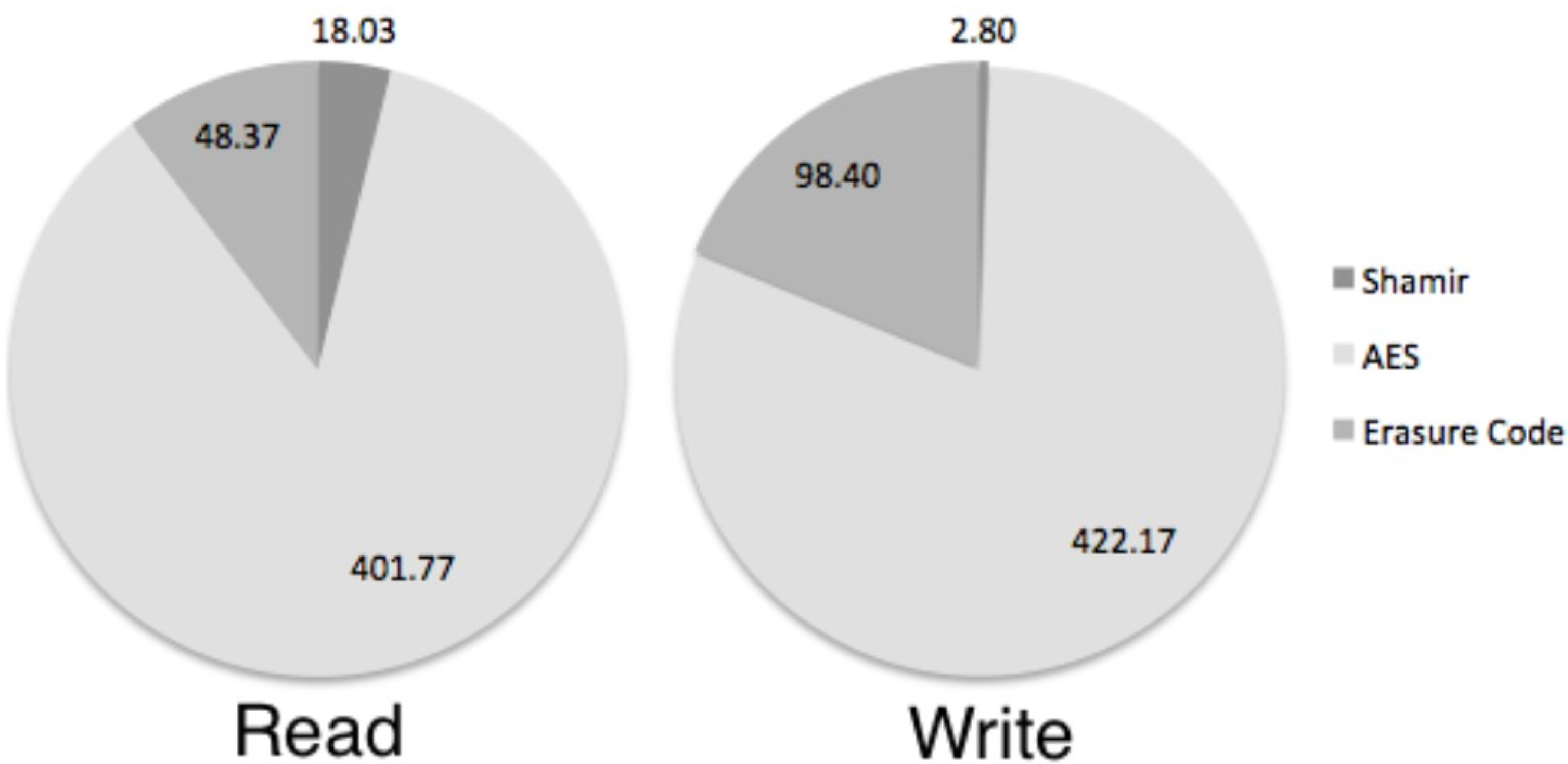
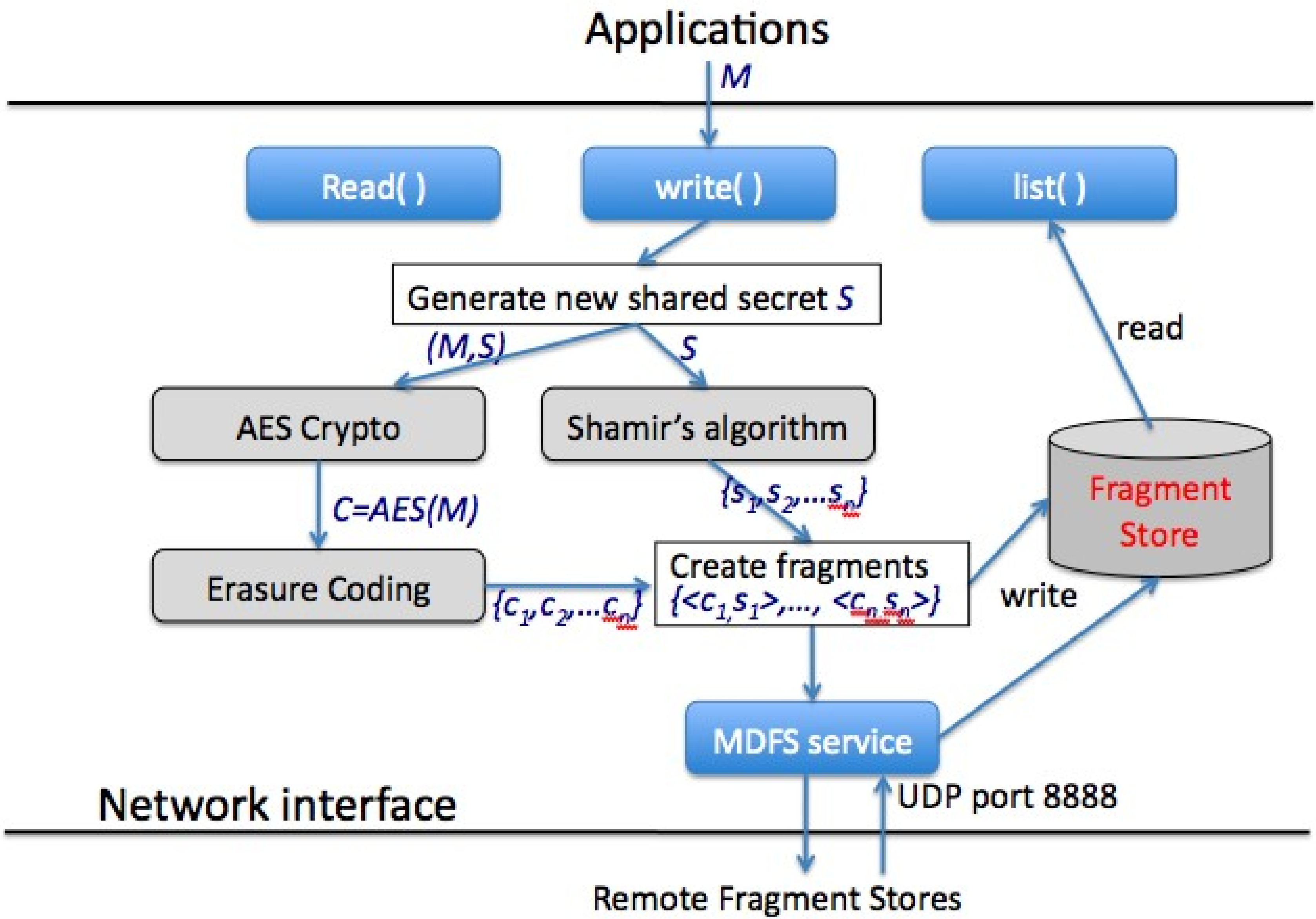


Fig. 4. Average Executive Times (ms) for 1MB file



Applications
M
Read( )
write( )
list( )
Generate new shared secret S
(M,S)
S
read
AES Crypto
Shamir's algorithm
Fragment Store
C=AES(M)
{s1,s2,...sn}
Erasure Coding
Create fragments
{<c1,s1>,..., <cn,sn>}
write
{c1,c2,...cn}
MDFS service
Network interface
UDP port 8888
Remote Fragment Stores